

```
In [2]: import cv2
import numpy as np
import matplotlib.pyplot as plt
```

```
In [3]: img = cv2.imread("/content/drive/My Drive/Colab Notebooks/Imagem /city.png")
plt.imshow(img)
```

Out[3]: <matplotlib.image.AxesImage at 0x7f6d5469fc50>



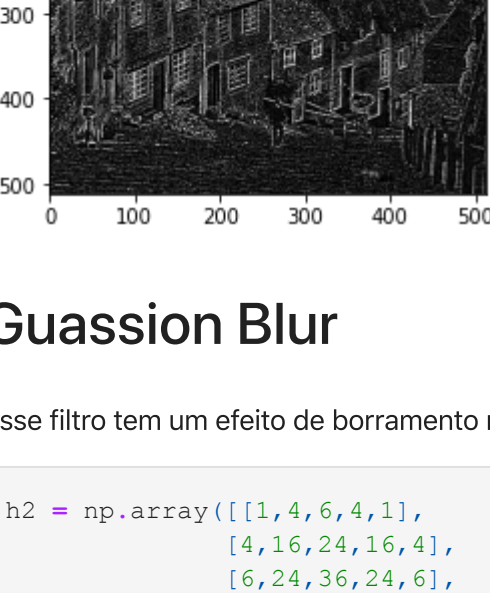
```
In [5]: def convolucao(image, kernel):
return cv2.filter2D(image,-1,cv2.flip(kernel,-1))
```

Laplaciano de Gaussiana

Esse filtro teve um efeito de realce das bordas da imagem

```
In [ ]: h1 = np.array([[0,0,-1,0,0],
[0,-1,-2,-1,0],
[-1,-2,16,-2,-1],
[0,-1,-2,-1,0],
[0,0,-1,0,0]])
imgH1 = convolucao(img,h1)
plt.imshow(imgH1)
```

Out[]: <matplotlib.image.AxesImage at 0x7f41abd5acc0>

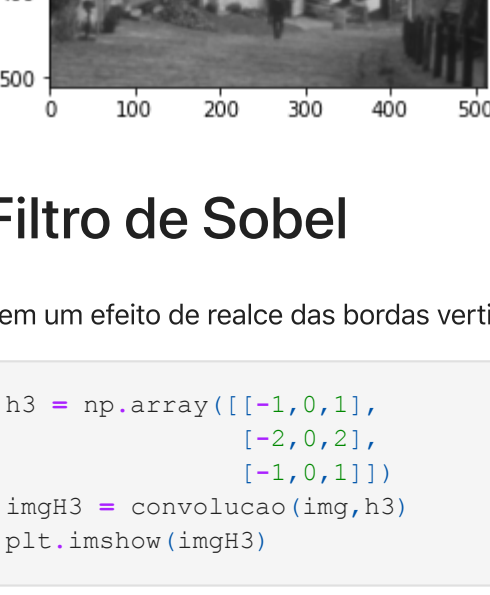


Guassion Blur

Esse filtro tem um efeito de borramento mais focalizado na imagem

```
In [ ]: h2 = np.array([[1,4,6,4,1],
[4,16,24,16,4],
[6,24,36,24,6],
[4,16,24,16,4],
[1,4,6,4,1]]) * (1/256)
imgH2 = convolucao(img,h2)
plt.imshow(imgH2)
```

Out[]: <matplotlib.image.AxesImage at 0x7f41abd39c18>

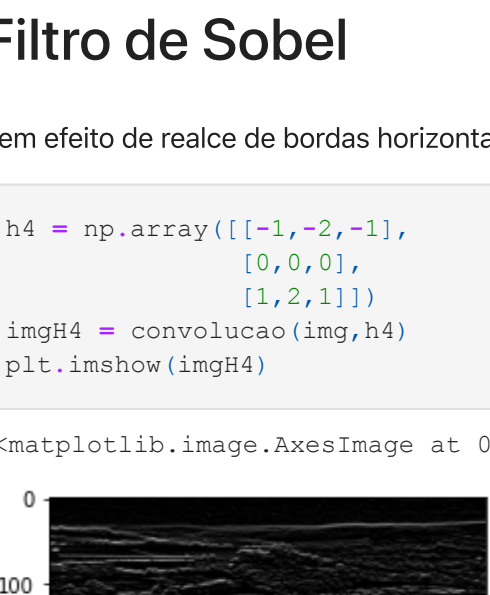


Filtro de Sobel

Tem um efeito de realce das bordas verticais da imagem

```
In [ ]: h3 = np.array([[[-1,0,1],
[-2,0,2],
[-1,0,1]])
imgH3 = convolucao(img,h3)
plt.imshow(imgH3)
```

Out[]: <matplotlib.image.AxesImage at 0x7f41abc9a8d0>

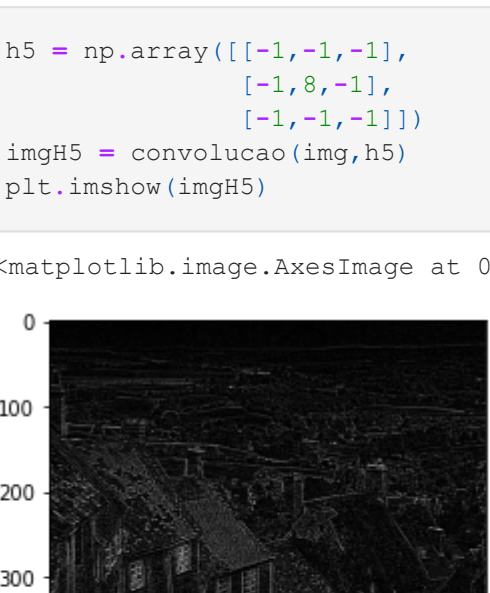


Filtro de Sobel

Tem efeito de realce de bordas horizontais da imagem

```
In [ ]: h4 = np.array([[[-1,-2,-1],
[0,0,0],
[1,2,1]])
imgH4 = convolucao(img,h4)
plt.imshow(imgH4)
```

Out[]: <matplotlib.image.AxesImage at 0x7f41abc789e8>

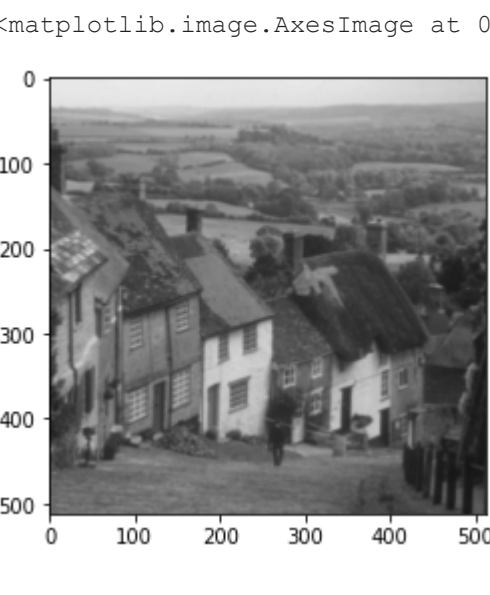


Filtro de Passa Alta

Esse filtro teve um efeito de realce das bordas da imagem

```
In [ ]: h5 = np.array([[[-1,-1,-1],
[-1,8,-1],
[-1,-1,-1]])
imgH5 = convolucao(img,h5)
plt.imshow(imgH5)
```

Out[]: <matplotlib.image.AxesImage at 0x7f41abc56c18>

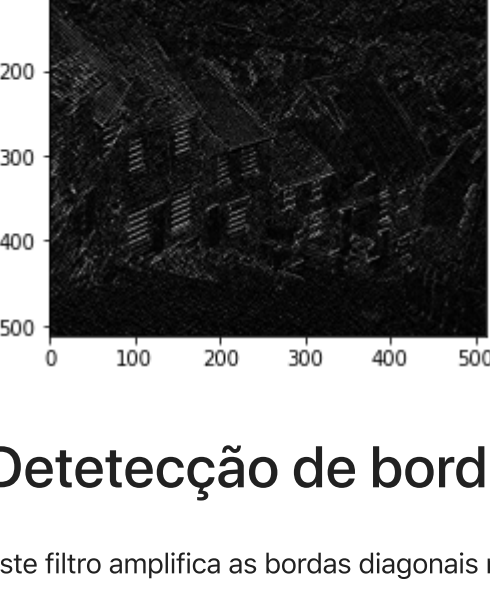


Filtro de Caixa

Esse filtro teve um efeito de suavização de detalhes na imagem

```
In [ ]: h6 = np.ones((3,3,float)) * (1/9)
imgH6 = convolucao(img,h6)
plt.imshow(imgH6)
```

Out[]: <matplotlib.image.AxesImage at 0x7f41abba390>

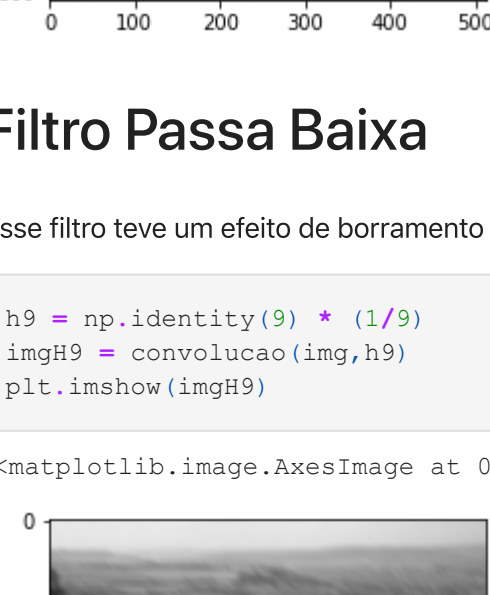


Detetecção de bordas Diagonais

Este filtro amplifica as bordas diagonais no sentido da diagonal secundária da imagem

```
In [ ]: h7 = np.array([[[-1,-1,2],
[-1,2,-1],
[2,-1,-1]])
imgH7 = convolucao(img,h7)
plt.imshow(imgH7)
```

Out[]: <matplotlib.image.AxesImage at 0x7f41ab89d2e8>

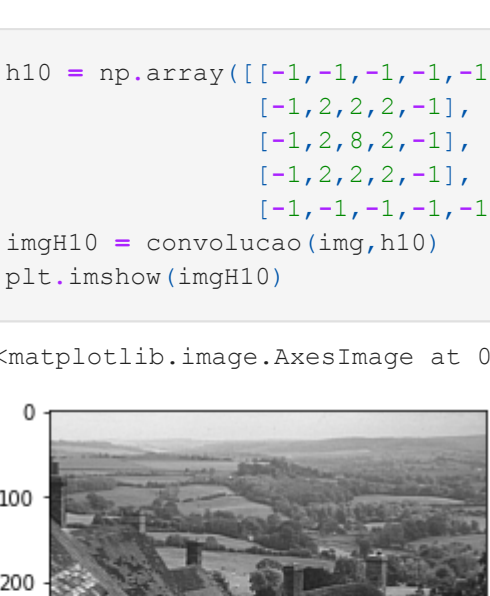


Detetecção de bordas Diagonais

Este filtro amplifica as bordas diagonais no sentido da diagonal principal da imagem

```
In [ ]: h8 = np.array([[2,-1,-1],
[-1,2,-1],
[-1,-1,2]])
imgH8 = convolucao(img,h8)
plt.imshow(imgH8)
```

Out[]: <matplotlib.image.AxesImage at 0x7f41abaeceb8>

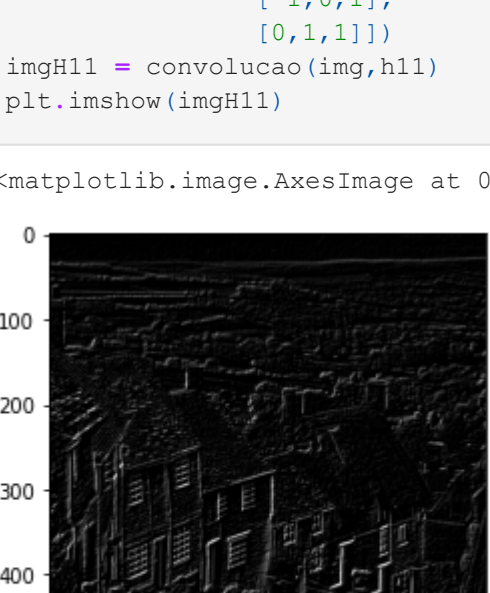


Filtro Passa Baixa

Esse filtro teve um efeito de borramento mais uniforme na imagem

```
In [9]: h9 = np.identity(9) * (1/9)
imgH9 = convolucao(img,h9)
plt.imshow(imgH9)
```

Out[9]: <matplotlib.image.AxesImage at 0x7f6d528585f8>



Foco

Esse filtro teve um efeito na imagem de dar maior nitidez

```
In [ ]: h10 = np.array([[[-1,-1,-1,-1,-1],
[-1,2,2,2,-1],
[-1,2,8,2,-1],
[-1,2,2,2,-1],
[-1,-1,-1,-1,-1]]) * (1/8)
imgH10 = convolucao(img,h10)
plt.imshow(imgH10)
```

Out[]: <matplotlib.image.AxesImage at 0x7f41aba28eb8>



Passa Alta

Esse filtro teve um efeito de realce das bordas da imagem

```
In [ ]: h11 = np.array([[[-1,-1,0],
[-1,0,1],
[0,1,1]])
imgH11 = convolucao(img,h11)
plt.imshow(imgH11)
```

Out[]: <matplotlib.image.AxesImage at 0x7f41aba0c390>

Combinação de filtros

Fazendo a combinação das imagens passadas por filtros de sobel, um noi sentido vertical e outro sentido vertical, temos uma imagem com um realce combinado das bordas verticais e horizontais

```
In [ ]: imgH12 = np.sqrt(np.power(np.float32(imgH3),2) + np.power(np.float32(imgH4),2))
plt.imshow(np.uint(imgH12))
```

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integer s).

Out[]: <matplotlib.image.AxesImage at 0x7f41ab95ff98>

